

ATIC-TISI

07 July 2000

MEMORANDUM FOR AIMS-PC USERS

SUBJECT: Software Version Description (SVD) for Installation Support Module (ISM) Automated Instructional Management System – Personal Computer (AIMS-PC) Interim Change Package (ICP) F5D-A59-02-02

1. Enclosed is ICP F5D-A59-02-02 that provides a formal release of AIMS-PC.
2. This memorandum certifies that the ISM subject software has been thoroughly tested, is considered a quality product, and is authorized for release to a production environment.
3. This change does not affect date calculations or computations. It has no Y2K ramifications.
4. Security Implications: None.

FOR THE COMMANDER:

Encl.

Chief Training Management Systems
ATISD

**SOFTWARE VERSION DESCRIPTION
(SVD)**

for

**AUTOMATED INSTRUCTIONAL MANAGEMENT SYSTEM
– PERSONAL COMPUTER
(AIMS-PC)**

ICP-F5D-A59-02-02



7 July 2000

TABLE OF CONTENTS

1	SCOPE.....	1
1.1	Identification.....	1
1.2	System Overview.....	1
1.3	Document Overview	1
2	REFERENCE DESCRIPTION	2
3	VERSION DESCRIPTION	2
3.1	Inventory of Materials Released	2
3.1.1	Media.....	2
3.1.2	Documentation.....	2
3.2	Inventory of Software Contents.....	2
3.2.1	Application Software Inventory.....	2
3.2.2	Database Software Inventory.....	2
3.3	Changes Installed.....	2
3.3.1	Corrector Descriptions – Technical	3
3.3.2	Installation Effective Date	3
3.3.3	Security Implication.....	3
3.3.4	Special Installation Instructions	3
3.3.5	Point of Contact.....	6
3.4	Possible Problems and Known Errors	6
4	NOTES	6
4.1	Runtime Environment	6
APPENDIX A: THE SCRIPTS	A-1	
CRINDEXES.SCR	SCRIPT:.....	A-2
DEFRAG_INDEXES.SH	SCRIPT:.....	A-4
DISABLE_PK.SCR	A-9
DISABLE_FK.SCR	A-10
DROP_INDEX.SCR	A-11
ENABLE_PK.SCR	A-12
ENABLE_FK.SCR	A-13
INDEX_RPT.SCR	A-14
LVST_0202.SH	A-16
RM_INDX_STAT.SCR	A-22
RM_STAT.SH	A-23
RM_TAB_STAT.SCR	A-26
SET_OPTIMIZER.SH	A-27
TS_MAP.SCR	A-29

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

1 SCOPE

This Software Version Description (SVD) describes the contents and installation of the F5D-A59-02-02 release of the Automated Instructional Management System – Personal Computer (AIMS-PC) for the Installation Support Modules (ISM) to be fielded. The Data Item Description (DID) for this document was developed to accommodate a remote installation of the product.

1.1 Identification

The hardware platform running the AIMS-PC is on Windows NT and Windows 95/98 on Personal Computers. The database is running on a UNIX platform networked AIX operating systems.

1.2 System Overview

AIMS-PC is an automatic, interactive training information management system for U.S. Army schools and training centers. This application provides the means to conduct, evaluate, and assess training to provide accurate information for making training decisions.

- a. The AIMS-PC system supports training registration and enrollment of students, test scoring, academic records, graduation, academic evaluation reports, and management reports. AIMS-PC provides class, course, lesson, student, unit, and group management functions. The AIMS-PC application provides methods to import data from legacy applications.
- b. An Ethernet Local Area Network (LAN) handles communications between users at a site. The main application software is on Personal Computers (PC). The data files are on an IBM RS/6000 Model R-24 running AIX 4.2.1 and serving as a data server. Data is passed between the PC's and the data server over the LAN. LANs in separate buildings on an installation are interconnected via intelligent hubs using a fiber backbone. Communications between sites and installations is achieved over a Wide Area Network (WAN). The LANs are connected to the WAN through a router. This allows transfer of information (data and files) and electronic mail to be sent between ISM sites, including Major Army Commands (MACOMs), the Information System Management and Control Center (ISMCC), and Contingency Control Facilities (CCF), and other installations, at dedicated transmission rates up to 1.544 Mbps. The ISM system supports the reuse of existing network and router capabilities such as Defense Data Network (DDN) as backup in case of failure. It also enables ISM to interface with the Army tactical networks.
- c. This system can receive from legacy systems.
- d. The AIMS-PC system receives information from the following legacy systems:
 - ? Army Training Requirements and Resources System (ATRRS)
 - ? Reception Battalion Automated Support System (RECBASS)
 - ? Program of Instruction Management Module (POIMM).

1.3 Document Overview

This document contains three major sections and appendices. Section 1 contains introductory/summary material about the application being delivered. Section 2 contains those materials necessary for

installation and checkout of the deliverable item. Section 3 contains notes that may be applicable to the installation or operation of the deliverable item. The appendices contain the shell script.

2 REFERENCE DESCRIPTION

This paragraph has been tailored out.

3 VERSION DESCRIPTION

This section contains the inventory of the deliverable and the instructions necessary for its installation. Where appropriate, certain materials are contained in and referenced to the appendices.

3.1 Inventory of Materials Released

The following subparagraphs list by identifying numbers, titles, dates, version numbers, and release numbers all physical media and associated documentation that makes up the version being released.

3.1.1 Media

This delivery of the AIMS-PC system to be fielded was done remotely using File Transfer.

3.1.2 Documentation

This paragraph has been tailored out.

3.2 Inventory of Software Contents

The following paragraphs list by identifying numbers, titles, dates, version numbers, and release numbers all computer files that make up the version being released. (Note: All Users: This ICP is to be implemented on ISM data servers, and should only be implemented for AIMS-PC database instances.)

3.2.1 Application Software Inventory

This paragraph tailored out.

3.2.2 Database Software Inventory

PROGRAM	VERSION	UPDATE
F5D-A59-02-02.tar.Z	4.3	20 June 2000
F5D-A59-02-02.readme	4.3	21 June 2000

3.3 Changes Installed

This ICP performs defragmentation of the AIMS-PC database indexes, sets optimization to rules-based, and removes table statistics previously used by the cost-based optimizer.

3.3.1 Corrector Descriptions – Technical

1. ECP-S F5D-R103-263 (STARS-8070)
 - A. CYCLES: None.
 - B. PROGRAM: AIMS-PC
 - C. PROBLEM: Problems generating Class Roster report. Databases were running with cost based optimizer with old staticis, databases were also fragmented.
 - D. RESOLUTION: Change from cost-based optimizer to rule-based optimizer, remove the statistics associated with the cost-based optimizer and defragment database indexes.
 - E. ICP: F5D-A59-02-02

3.3.2 Installation Effective Date

This change package must implementation within 30 days of release.

3.3.3 Security Implication

None

3.3.4 Special Installation Instructions

Data Server Installation

The tar file is in the **release/F5D-A59-02-02** directory on server **155.217.50.29**.

1. Start FTP program. Log onto 155.217.50.29.
2. On the remote system. Change to the release/F5D-A59-02-02 directory
3. Set binary mode on.
4. Using FTP command get files F5D-A59-02-02.tar.Z, F5D-A59-02-02.readme, and F5D-A59-02-02SVD.doc
5. To extract the tar.Z file on a UNIX platform, follow the instructions in the F5D-A59-02-02.readme file. The readme file may be opened with any UNIX text editor.

ANSOC receives the baseline with change package number. This change package contains the following files:

Quantity	File ID	Version	Change Package No.
1	F5D-A59-02-02.tar.Z	4.3	F5D-A59-02-02
1	F5D-A59-02-02.readme	4.3	F5D-A59-02-02
1	F5D-A59-02-02SVD.doc	4.3	F5D-A59-02-02

F5D-A59-02-02.readme

Ensure that all master file backups have been completed prior to loading this package and running the scripts.

The following steps need to be done for each AIMSPC and TRAINING database instance.

1. Make sure all other users are off the database before starting to run this script.

NOTE: This script will shutdown and startup the database

2. For the instance (ORACLE_SID) this script is to be run against, make sure the directories

```
$ORACLE_HOME/dbs  
/Cots/oracle/admin/<ORACLE_SID>/pfile
```

will allow files to be created in them and that the file init<ORACLE_SID>.ora in these two directories are writeable

NOTE: ORACLE_SID is the oracle instance the LVST script is to be run against

3. Make sure you are the UNIX user oracle
4. Change to the directory /Cots/oracle/admin/<ORACLE_SID>/scripts
5. At the Unix command line, execute the command:

```
mkdir lvst_0202
```

NOTE: directory name is lower case and includes an underscore (_)

6. Execute the following commands:

```
cd /Cots/oracle/admin/<ORACLE_SID>/scripts/lvst_0202  
uncompress -c <PATH_COMPRESS_TAR_FILE>/F5D-A59-02-02.tar.Z | tar xvfp -
```

This will uncompress and untar the files

NOTE: PATH_COMPRESS_TAR_FILE is the location where the file F5D-A59-02-02.tar.Z resides

7. Run the script lvst_0202.sh from the /Cots/oracle/admin/<ORACLE_SID>/scripts/lvst_0202 directory as follows from the Unix command line:

```
cd /Cots/oracle/admin/<ORACLE_SID>/scirpts/lvst_0202  
./lvst_0202.sh <ORACLE_SID>
```

where <ORACLE_SID> is the oracle instance the script is to be run against

- a. The script lvst_0202.sh will display the Oracle database instance you entered on the command line and ask you if the instance is correct.

If the instance is correct type:

y

If the instance is incorrect type:

n

The script will terminate. Now rerun the script with the correct instance.

- b. The script lvst_0202.sh will then ask for the Oracle system password. Enter the Oracle system password. The Oracle system password will not be echoed.
- c. While the script lvst_0202.sh is running, it will echo what it is doing. For example:

```
Starting script  
Disable other users from logging onto <ORACLE_SID>  
Remove the statistics  
Set optimizer mode to Rule  
Shutting down instance <ORACLE_SID>  
Sleep some - so things will settle out  
Startup the database  
Allow other users to log onto the <ORACLE_SID>  
Successfull completion of lvst_0202
```

- d. When the script is complete the following messasge will be displayed:

Successful completion of lvst_0202

8. Restart the Oracle listener if it is not up. At the Unix command line type:

lsnrctl reload

9. Now check the log file lvst_0202.log

The counts for the number of:

? Indexes

- ? Primary Unique Keys Enabled and Disabled
- ? Foreign Keys Enabled and Disabled
- ? Constraints disabled

should be the same at the start and end of the log file

The following error messages associated with the drop index statements can be ignored.

ORA-01418: specified index does not exist

3.3.5 Point of Contact

If there are any problems or questions with the installation or the contents of this package, call the Army Network and Systems Operator Center (ANSOC), Ft. Huachuca: DSN 879-6798 or 1-800-305-3036. If there are any application or technical problems, Army Training Support Center (ATSC), Ft Eustis: DSN 927-4881 ext 258 or 1-800-ASK-ATSC

3.4 Possible Problems and Known Errors

None known at this time.

4 NOTES

4.1 Runtime Environment

The AIMS-PC System has been certified to run in a distributed PC, data server environment. The application user display device is a Personal Computer (PC).

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

APPENDIX A: THE SCRIPTS

APPENDIX A: THE SCRIPTS

The following script files are contained in the F5D-A59-02-02.tar.Z file:

? crindexes.scr	3038K	Jun 19 2000
? defrag_indexes.sh	6244K	Jun 19 2000
? disable_fk.scr	560K	Jun 19 2000
? disable_pk.scr	576K	Jun 19 2000
? drop_index.scr	311K	Jun 19 2000
? enable_fk.scr	406K	Jun 19 2000
? enable_pk.scr	413K	Jun 19 2000
? index_rpt.scr	1843K	Jun 19 2000
? lvst_0202.sh	6759K	Jun 19 2000
? rm_idx_stat.scr	324K	Jun 19 2000
? rm_stat.sh	2795	Jun 19 2000
? rm_tab_stat.scr	1267	Jun 19 2000
? set_optimizer.sh	1993K	Jun 19 2000
? tsmap.scr	1049K	Jun 19 2000

CRINDEXES.SCR SCRIPT:

REM

REM Obtain this script from Internet site

REM

set linesize 80;

set pagesize 10000;

set long 50;

set echo on;

set heading off;

/* ***** */

/* */

/* Script to spool a listing of all create */

/* statements required to rebuild AIMSPC */

/* indexes. */

/* Contents: Table name, col definitions and */

/* storage parameters. */

/* */

/* */

/* ***** */

set termout off;

set echo off;

set feedback off;

col dummy noprint format a1;

col dummy2 noprint format a1;

col index_name noprint format a1;

col command format a80;

/* ***** */

/* Generate create index commands */

/* ***** */

select 'create'||decode(UNIQUENESS, 'UNIQUE', ' UNIQUE', NULL)||'

index ''||owner||'.'||index_name||'

on '

||table_owner||'.'||table_name|| '(' command,

1 dummy,

index_name,

0 dummy2

from all_indexes

where table_owner = 'AIMSPC'

UNION

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
select '||decode(column_position,1,null,')||column_name command,
       2                      dummy,
       index_name,
       column_position          dummy2
  from all_ind_columns
 where table_owner = 'AIMSPC' AND
       all_ind_columns.index_name in
         (select index_name
            from all_indexes
           where table_owner = 'AIMSPC' )
UNION
 select ' )'                  command,
       3                      dummy,
       index_name,
       0                      dummy2
  from all_ind_columns
 where table_owner = 'AIMSPC' AND
       all_ind_columns.index_name in
         (select index_name
            from all_indexes
           where table_owner = 'AIMSPC')
UNION
 select ' pctfree '||pct_free||
       initrans '||ini_trans||
       maxtrans '||max_trans||
       storage (initial '||initial_extent ||
                next     '||next_extent ||
                minextents '||min_extents ||
                maxextents '||max_extents ||
                pctincrease '||pct_increase ||
                freelists '||freelists ||
                freelist groups '||freelist_groups ||
        ) tablespace '||tablespace_name||' unrecoverable;'   command,
       4                      dummy,
       index_name,
       0                      dummy2
  from all_indexes
 where table_owner = 'AIMSPC'
order by 3,2,4;
```

spool cr_index.sql

/

spool off;

set heading on;

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
set pagesize 100;
set termout on;
```

DEFRAG_INDEXES.SH SCRIPT:

```
#!/bin/sh
#
# File Name: defrag_indexes
# Author: XXXX
# Date: 7 May 2000
# Modified by XXXX
#
# Purpose:
#
#   Defrags AIMSPC indexes.
#
#
# Usage 1: defrag_indexes <instance name>
# Usage 2: defrag_indexes <instance name> <userid> <password>
#
# This script will perform the following functions:
# 1. Validate command line arguments checking for valid instance name in the
#    /etc/oratab file.
# 2. Prompts for userid and password if no userid and password entered
#    on the command line
# 3. Prepares index, primary key and foreign key scripts to use in rebuild.
# 4. Displays before and after counts of indexes, primary and foreign keys.
#####
#
trap 'exit' 1 2 3
case $ORACLE_TRACE in
    T)set -x ;;
esac
#####
#
alias rmit=rm

args=$#
if [ ${args} -lt 1 ]
then
    echo "Invalid number of arguments"
    echo "Usage: defrag_indexes.sh <instance> "
    exit 1
fi
#####
#
ORATAB=/etc/oratab
temp_sid=$1
userid=$2
password=$3
ORACLE_HOME= ""
ORACLE_SID= ""
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
hostname=`exec hostname`  
  
# Set path if path not set (if called from /etc/rc)  
case $PATH in  
    "") PATH=/bin:/usr/bin:/etc  
        export PATH ;;  
esac  
  
# Find Matching ORACLE_SID in the ORATAB file to set the ORACLE_HOME and PATH  
cat $ORATAB | while read LINE  
do  
    case $LINE in  
        \#*);; #comment-line in oratab  
        *)  
            ORACLE_SID=`echo $LINE | awk -F: '{print $1}' -`  
            if [ "${ORACLE_SID}" = "${temp_sid}" ]  
            then  
                break  
            else  
                ORACLE_SID=""  
            fi  
    esac  
done  
  
# Just In case There is no carriage return after the last line - do it again  
ORACLE_SID=`echo $LINE | awk -F: '{print $1}' -`  
if [ "${ORACLE_SID}" = "${temp_sid}" ]  
then  
    export ORACLE_SID  
    ORACLE_HOME=`echo $LINE | awk -F: '{print $2}' -`  
    export ORACLE_HOME  
    # Put $ORACLE_HOME/bin into PATH and export.  
    PATH=$ORACLE_HOME/bin:/bin:/usr/bin:/etc ; export PATH  
else  
    echo "Error Executing $PWD/defrag_indexes\nInvalid Parameter ORACLE_SID -  
${temp_sid}"  
    exit 1  
fi  
  
echo "\n`date` - Host: ${hostname}\n\ndefrag Indexes on \"${ORACLE_SID}\"  
Instance..."  
echo ""  
if [[ $userid  = "" ]]  
then  
    echo "ENTER Oracle SYSTEM Id: \c"  
    read userid  
    export userid  
    stty -echo  
    echo ""  
    echo "ENTER password: \c"  
    read password  
    export password  
    echo "\n"  
    stty echo  
fi
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
#Test for valid userid/password
sqlplus -s ${userid}/${password}<<EOF
exit
exit
exit
exit
exit
exit
EOF
if [ $? != 0 ]
then
echo "Entered Invalid UserId/Password"
exit 1
fi

echo "\nExtract Existing Index Info for rebuild ....\n"
sqlplus -s ${userid}/${password} <<EOF
@tsmap.scr sbis_aimspc_index1
EOF

sqlplus -s ${userid}/${password} <<EOF
@index_rpt.scr  sbis_aimspc_index1
EOF

sqlplus -s ${userid}/${password} <<EOF
@crindexes.scr
EOF

# Remove blank lines from index creation file
#
mv cr_index.sql cr_index.tmp
sed -e '/^ *$/d' cr_index.tmp > cr_index.sql
rm cr_index.tmp

sqlplus -s ${userid}/${password}<<EOF
set pause off
set heading off
select 'Capture Counts Prior To Performing Defrag' from dual;
set heading on
set feedback off
spool counts_sbis_aimspc_index1
select count(*) "# Of Indexes" from all_indexes where owner = 'AIMSPC';
select count(*) "# Of Prim/Unique Keys Enabled" from all_constraints
  where constraint_type in ('P','U') and status = 'ENABLED';
select count(*) "# Of Prim/Unique Keys Disabled" from all_constraints
  where constraint_type in ('P','U') and status = 'DISABLED';
select count(*) "# Of Foreign Keys Enabled" from all_constraints
  where constraint_type = 'R' and status = 'ENABLED';
select count(*) "# Of Foreign Keys Disabled" from all_constraints
  where constraint_type = 'R' and status = 'DISABLED';
select constraint_name "Constraint", table_name "Table",status from
all_constraints where status='DISABLED';
set feedback on
spool off
@disable_fk.scr
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
@disable_pk.scr
@drop_index.scr
@Enable_pk.scr
@Enable_fk.scr
exit
EOF

prog_err=$?

if [[ ${prog_err} != 0 ]]
then
    echo "`date` - Defrag Indexes Failed $prog_err"
    exit 1
else
    echo "\nChecking for errors in defrag preparation scripts...\n"
    for FILE in `find . -print`
    do
        case $FILE in
            *.sql)

                err=`grep -ic "Error" $FILE`
                if [ ${err} != 0 ]
                then
                    echo "`date` - Defrag Indexes $FILE has errors"
                    echo "Scripts for Defrag Preparation Failed - Defrag Not Started"
                    echo "Database remains in present state."
                    exit 1
                fi
            esac
        done
    fi
echo "No Errors - Defrag Preparation Scripts Complete.\n"
echo "\nStarting Defrag of AIMSPC Indexes...\n"
sqlplus -s ${userid}/${password}<<EOF
@Enable_fk
@Enable_pk
@drop_index
alter tablespace sbis_aimspc_index1 coalesce;
alter tablespace sbis_aimspc_data1 coalesce;
@Enable_pk
@Enable_fk
set heading off
set feedback off
select 'Counts After Defrag' from dual;
set heading on
spool counts_SBIS_AIMSPC_INDEX1
select count(*) "# Of Indexes" from all_indexes where owner = 'AIMSPC';
select count(*) "# Of Indexes" from all_indexes where owner = 'AIMSPC';
select count(*) "# Of Prim/Unique Keys Enabled" from all_constraints
    where constraint_type in ('P','U') and status = 'ENABLED';
select count(*) "# Of Prim/Unique Keys Disabled" from all_constraints
    where constraint_type in ('P','U') and status = 'DISABLED';
select count(*) "# Of Foreign Keys Enabled" from all_constraints
    where constraint_type = 'R' and status = 'ENABLED';
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
select count(*) "# Of Foreign Keys Disabled" from all_constraints
  where constraint_type = 'R' and status = 'DISABLED';
select constraint_name "Constraint", table_name,status "Table" from
all_constraints where status='DISABLED';
spool off
set heading off
select 'Contact AIMSPC Analysts If Counts Prior to Defrag are Different than
After.' from dual;
@tsmap.scr SBIS_AIMSPC_INDEX1
EOF

sqlplus -s ${userid}/${password} <<EOF
@index_rpt.scr  SBIS_AIMSPC_INDEX1
EOF
echo "\n AIMSPC defrag Complete\n"
```

DISABLE_PK.SCR

```
set term off
set wrap on
set verify off
set heading off
set pages 9999
set pause off
set feedback off
set lines 100
spool disable_pk.sql
select 'alter table '||owner||'.'||table_name|| ' disable constraint '
|| constraint_name||';' from all_constraints where
owner = 'AIMSPC' and constraint_type in ('P','U') and status = 'ENABLED'
/
select 'select constraint_name,status from all_constraints where owner =
''AIMSPC'' and constraint_type in (''P'', ''U'') and status = ''ENABLED''; '
from dual
/
spool off
set term on
select 'Created disable_pk.sql...' from dual;
set term off
```

DISABLE_FK.SCR

```
set term off
set wrap on
set verify off
set heading off
set pages 9999
set pause off
set feedback off
set lines 100
spool disable_fk.sql
select 'alter table '||owner||'.'||table_name|| ' disable constraint ' 
|| constraint_name||';' from all_constraints where
owner = 'AIMSPC' and constraint_type = 'R' and status = 'ENABLED'
/
select 'select constraint_name,status from all_constraints where owner = 
''AIMSPC'' and constraint_type = ''R'' and status = ''ENABLED'';' from dual
/
spool off
set term on
select 'Created disable_fk.sql...' from dual;
set term off
```

DROP_INDEX.SCR

```
set term off
set wrap on
set verify off
set heading off
set pages 9999
set pause off
set feedback off
set lines 90
spool drop_index.sql
select 'drop index '||owner||'.'||index_name|| ' ; '
from all_indexes where owner = 'AIMSPC'
/
spool off
set term on
select 'Created drop_index.sql...' from dual;
set term off
```

ENABLE_PK.SCR

```
set term off
set wrap on
set verify off
set heading off
set pages 9999
set pause off
set feedback off
set lines 100
spool enable_pk.sql
select 'alter table '||owner||'.'||table_name|| ' enable constraint '
|| constraint_name||';' from all_constraints where
owner = 'AIMSPC' and constraint_type in ('P','U') and status = 'ENABLED'
/
spool off
set term on
select 'Created enable_pk.sql...' from dual;
set term off
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

ENABLE_FK.SCR

```
set term off
set wrap on
set verify off
set heading off
set pages 9999
set pause off
set feedback off
set lines 100
spool enable_fk.sql
select 'alter table '''||owner||'.'||table_name|| ' enable constraint '
|| constraint_name||';' from all_constraints where
owner = 'AIMSPC' and constraint_type = 'R' and status = 'ENABLED'
/
spool off
set term on
select 'Created enable_fk.sql...' from dual;
set term off
```

INDEX_RPT.SCR

```
REM
REM index_rpt.scr
REM -----
REM This report gives information about the indexes associated
REM with the aimspc user
REM
set echo off
set termout off
set feedback off
set pause off
set verify off
set newpage 0
set linesize 80
set pagesize 60

column owner format a10 word_wrapped
column index_name format a20 word_wrapped
column table_name format a15 word_wrapped
column uniqueness format a10 word_wrapped
column uniqueness heading 'Unique'
column tablespace_name format a15 word_wrapped
column ini_trans format 99999
column ini_trans heading 'ITRNS'
column max_trans format 99999
column max_trans heading 'MTRNS'
column initial_extent format 9999999
column initial_extent heading 'Init_ext'
column max_extents format 9999999
column max_extents heading 'Max_ext'
column pct_increase format 999
column pct_increase heading 'Pct'
column freelists format 999
column freelists heading 'Free'
column freelist_groups format 999
column freelist_groups heading 'Grp'

spool index_rpt_&1
ttitle center 'I N D E X   M A P' skip 1 -
    center (&1) skip 1 -
    center '=====' right 'PAGE:' -
    format 999 sql.pno skip 2

select owner, index_name, table_name, uniqueness,
       tablespace_name
from all_indexes
where owner = 'AIMSPC'
order by table_name, index_name
;

select owner, index_name,
       ini_trans, max_trans, initial_extent,
       max_extents, pct_increase, freelists, freelist_groups
from all_indexes
where owner = 'AIMSPC'
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
order by table_name, index_name
;

column index_owner format a10
column index_name format a20 word_wrapped
column column_name format a15 word_wrapped
column table_name format a15 word_wrapped

select index_owner, index_name, table_name,
       column_name
  from dba_ind_columns
 where index_owner = 'AIMSPC'
 order by index_name, table_name, column_name
;

spool off
```

LVST_0202.SH

```
#!/usr/bin/ksh
#
# lvst_0202.sh
#
# Purpose:
# To perform the following tuning functions:
# 1 - Defragment the database indexes
# 2 - Remove statistics from the aimspc tables
# 3 - Change the optimizer mode to rule
#
# Usage 1: lvst_0202 <instance_name>
# 1 - database instance
#
# This script will perform the following fuctions:
# 1. Validate command line argument and check for valid instance name in
#    the /etc/oratab file
# 2. Check that this script is located in the correct directory, if script
#    is not located in the correct directory, script will terminate
# 3. Ask the user for Oracle system password, if incorrect Oracle system
#    password entered, script will terminate
#
# Setup of log file
#
LOG=lvst_0202.log
export LOG
echo "\n`date` Starting script"
echo "\n`date` Starting script" >> $LOG
echo " "
echo " " >> $LOG

#
# Check for valid number of arguments passed
#
args=$#
if [ ${args} -ne 1 ]
then
    echo "\nInvalid number of arguments"
    echo "\nInvalid number of arguments" >> $LOG
    echo "Usage: lvst_0202.sh <instance> "
    echo "Usage: lvst_0202.sh <instance> " >> $LOG
    exit 1
fi

temp_sid=$1

#
# Check that user oracle is running the script
# Otherwise won't be able to get into svrmgrl without password
#
user=`whoami`
if [ "$user" != "oracle" ]
then
    echo "\nExecute Permission Denied"
    echo "\nExecute Permission Denied" >> $LOG
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
echo "Must execute as oracle"
echo "Must execute as oracle" >> $LOG
exit 1
fi

#
# Set path if path not set (if called from /etc/rc)
#
case $PATH in
  "") PATH=/bin:/usr/bin:/etc
      export PATH ;;
esac

#
# Find Matching ORACLE_SID in the ORATAB file to set the ORACLE_HOME and PATH
#
ORATAB=/etc/oratab
ORACLE_HOME=""
ORACLE_SID=""

cat $ORATAB | while read LINE
do
  case $LINE in
    *\#*);; #comment-line in oratab
    *)
      ORACLE_SID=`echo $LINE | awk -F: '{print $1}' -` 
      if [ "${ORACLE_SID}" = "${temp_sid}" ]
      then
        break
      else
        ORACLE_SID=""
      fi
    esac
done
#
# Just In case There is no carriage return after the last line - do it again
#
ORACLE_SID=`echo $LINE | awk -F: '{print $1}' -` 
if [ "${ORACLE_SID}" = "${temp_sid}" ]
then
  export ORACLE_SID

  ORACLE_HOME=`echo $LINE | awk -F: '{print $2}' -` 
  export ORACLE_HOME

  # Put $ORACLE_HOME/bin into PATH and export.
  PATH=$ORACLE_HOME/bin:/bin:/usr/bin:/etc
  export PATH
else
  echo "\nInvalid Parameter ORACLE_SID - ${temp_sid}"
  echo "\nInvalid Parameter ORACLE_SID - ${temp_sid}" >> $LOG
  exit 1
fi

echo "ORACLE_SID is $ORACLE_SID"
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
#  
# Verify with user that instance is correct  
#  
echo ""  
echo "Database instance is == $ORACLE_SID =="  
echo "Is this the correct instance [y/n] : \c"  
read answer  
if [ "$answer" != "y" ]; then  
    echo "\nUser entered incorrect database instance of $ORACLE_SID"  
    echo "\nUser entered incorrect database instance of $ORACLE_SID" >> $LOG  
    exit 1  
fi  
#  
# Check if this script is in correction location for instance  
#  
echo "PWD = $PWD"  
PART1=/Cots/oracle/admin  
export PART1  
SCRIPT_LOCATION=${PART1}/${ORACLE_SID}/scripts/lvst_0202  
export SCRIPT_LOCATION  
  
if [ "$SCRIPT_LOCATION" != "$PWD" ]  
then  
    echo "\nScript is located in wrong directory"  
    echo "\nScript is located in wrong directory" >> $LOG  
    echo "should be located in $SCRIPT_LOCATION"  
    echo "should be located in $SCRIPT_LOCATION" >> $LOG  
    echo "instead of $PWD"  
    echo "instead of $PWD" >> $LOG  
    exit 1  
fi  
  
#  
# Ask user for oracle system password  
#  
  
echo ""  
echo "Enter Oracle system password : \c"  
stty -echo  
read SYSTEM_PASSWORD  
stty echo  
export SYSTEM_PASSWORD  
echo ""  
  
#  
# Check if Oracle system password correctly entered  
#  
sqlplus -s <<ENDSQL  
system/${SYSTEM_PASSWORD}  
exit  
exit  
exit  
exit  
exit  
ENDSQL  
if [[ $? != 0 ]]
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
then
echo "\nInvalid Oracle system password entered"
echo "\nInvalid Oracle system password entered" >> $LOG
exit 1
fi

#-----
# Start main part of script
#-----
#
# disable new connections to the database
#
echo "\nDisable other users from logging onto $ORACLE_SID"
echo "\nDisable other users from logging onto $ORACLE_SID" >> $LOG
svrmgrl >>$LOG 2>>$LOG <<ENDSQLDBA
connect internal
alter system enable restricted session;
exit
ENDSQLDBA
if [ $? -ne 0 ]
then
    echo "\nDisabling connections to database $ORACLE_SID Failed."
    echo "\nDisabling connections to database $ORACLE_SID Failed." >> $LOG
    exit 1
fi

#
# Remove the statistics from the user tables in the database
#
echo "\nRemove the statistics"
echo "\nRemove the statistics" >> $LOG
./rm_stat.sh $ORACLE_SID system $SYSTEM_PASSWORD >> $LOG
if [[ $? != 0 ]]
then
    echo "\nFailed to remove statistics"
    echo "\nFailed to remove statistics" >> $LOG
    exit 1
fi

echo "\nDefrag the indexes"
echo "\nDefrag the indexes" >> $LOG
./defrag_indexes.sh $ORACLE_SID system $SYSTEM_PASSWORD >>$LOG 2>>$LOG
if [[ $? != 0 ]]
then
    echo "\nFailed to defrag indexes"
    echo "\nFailed to defrag indexes" >> $LOG
    exit 1
fi

echo "\nSet optimizer mode to Rule"
echo "\nSet optimizer mode to Rule" >> $LOG
./set_optimizer.sh $ORACLE_SID
if [[ $? != 0 ]]
then
    echo "\nFailed to set optimizer mode to rule"
    echo "\nFailed to set optimizer mode to rule" >> $LOG
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
    exit 1
fi

#
# Now after we have performed our work shutdown the database and
# restart it
#
echo "\nShutting down instance $ORACLE_SID"
echo "\nShutting down instance $ORACLE_SID" >> $LOG
svrmgrl >>$LOG 2>>$LOG <<ENDSQLDBA
connect internal
shutdown immediate
exit
ENDSQLDBA
if [ $? -ne 0 ]
then
    echo "\nUnable to shutdown database $ORACLE_SID Failed."
    echo "\nUnable to shutdown database $ORACLE_SID Failed." >> $LOG
    exit 1
fi

echo "\nSleep some - so things will settle out"
echo "\nSleep some - so things will settle out" >> $LOG
sleep 5

echo "\nStartup the database"
echo "\nStartup the database" >> $LOG
svrmgrl >>$LOG 2>>$LOG <<ENDSQLDBA
connect internal
startup
exit
ENDSQLDBA
if [ $? -ne 0 ]
then
    echo " "
    echo " " >> $LOG
    echo "Unable to startup database $ORACLE_SID Failed"
    echo "Unable to startup database $ORACLE_SID Failed" >> $LOG
    exit 1
fi

#
# Enable new connections to the database
#
echo "\nAllow other users to logging onto $ORACLE_SID"
svrmgrl >>$LOG 2>>$LOG <<ENDSQLDBA
connect internal
alter system disable restricted session;
exit
ENDSQLDBA
if [ $? -ne 0 ]
then
    echo "\nEnabling connections to database $ORACLE_SID Failed."
    echo "\nEnabling connections to database $ORACLE_SID Failed." >> $LOG
    exit 1
fi
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
#  
# Now we have successfully completed the script lvst_0202  
#  
echo "Successful completion of lvst_0202"  
echo "Successful completion of lvst_0202" >> $LOG  
exit 0
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

RM_INDX_STAT.SCR

```
REM
REM remove statistics from indexes
REM
set pagesize 0
set heading off
set feedback off
set termout off
set linesize 132

spool rm_indx_stat.sql

select 'analyze index ' || owner || '.' || index_name
|| ' delete statistics ;'
from dba_indexes
where owner in ('AIMSPC', 'COMMON', 'ENTMANAGER', 'OPS$ORACLE');

spool off
```

RM_STAT.SH

```
#!/usr/bin/ksh
#
# rm_stat.sh
#
# Remove statistics from tables and indexes
# If you are using OPTIMIZER_MODE = choose
# and you remove statistics from your tables and indexes
# then rule-based optimizer will be used instead of
# the cost-based optimizer
#
# Parameters
# 1 - instance name
# 2 - database user - usually system or administrator
# 3 - password - password required to log onto the database for database
#     user
# 4 - yes or no - if set to yes, then script will also remove statistics
#     from indexes, otherwise only table statistics will be removed
#     Default have is "no".
#
# NOTE: This script does minimal error checking - call
#       a driver script to do the error checking
#
echo "`date` - Starting rm_stat.sh"
echo "`date` - Starting rm_stat.sh" >> rm_stat.log
args=$#
if [ ${args} -lt 3 ]
then
    echo " "
    echo "Invalid number of arguments"
    echo "Invalid number of arguments" >> rm_stat.log
    echo "Usage: rm_stat.sh <instance> <db_user> <password>\n"
    echo "Usage: rm_stat.sh <instance> <db_user> <password>\n" >> rm_stat.log
    echo " "
    exit 1
fi
ORACLE_SID=$1
export $ORACLE_SID

userid=$2
password=$3
tmpindex=$4
index=${tmpindex:-no}

LOG=rm_stat.log

if [ "$ORACLE_HOME" = "" ]
then
    ORACLE_HOME="/Cots/oracle/product/7.3.3"
    export ORACLE_HOME
fi

#Test for valid userid/password
sqlplus -s ${userid}/${password}<<EOF
    exit
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
exit
exit
exit
exit
EOF
if [ $? != 0 ]
then
    echo "Entered Invalid UserId/Password\n"
    exit 1
fi

sqlplus -s ${userid}/${password} << EOF
@rm_tab_stat.scr
exit
EOF
if [[ $? != 0 ]]
then
    echo "\nFailed to generate script to remove table statistics"
    echo "\nFailed to generate script to remove table statistics" >> $LOG
    exit 1
fi

sqlplus -s ${userid}/${password} << EOF
@rm_tab_stat.sql
exit
EOF
if [[ $? != 0 ]]
then
    echo "\nFailed to remove user table statistics"
    echo "\nFailed to remove user table statistics" >> $LOG
    exit 1
fi

echo "\nFinished removing user table statistics "
echo "\nFinished removing user table statistics " >> $LOG

if [ "$index" = "no" ]
then
    echo "\nExiting rm_stat.sh "
    echo "\nExiting rm_stat.sh " >> $LOG
    exit 0
fi

sqlplus -s ${userid}/${password} << EOF
@rm_idx_stat.scr
exit
EOF
if [[ $? != 0 ]]
then
    echo "\nFailed to generate script to remove index statistics"
    echo "\nFailed to generate script to remove index statistics" >> $LOG
    exit 1
fi

sqlplus -s ${userid}/${password} << EOF
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
@rm_idx_stat.sql
exit
EOF
if [[ $? != 0 ]]
then
  echo "\nFailed to remove user index statistics"
  echo "\nFailed to remove user index statistics" >> $LOG
  exit 1
fi

echo "\nFinished removing user index statistics "
echo "\nFinished removing user index statistics " >> $LOG

echo "\nExiting rm_stat.sh "
echo "\nExiting rm_stat.sh " >> $LOG
exit 0
```

RM_TAB_STAT.SCR

```
REM
REM
REM rm_tab_stat.scr
REM -----
REM sql script which generates another sql script to delete the
REM statistics which have been generated by the ANALYZE command for
REM tables owned by users AIMSPC, COMMON, ENTMAGER, OPS$ORACLE
REM
REM Note: If you want to remove the statistics from the entire database
REM you must also remove the statistics on the indexes or rebuild the
REM indexes
REM
set pagesize 0
set heading off
set feedback off
set termout off
set linesize 80

spool rm_tab_stat.sql

REM
REM Remove statistics from tables
REM
select distinct 'ANALYZE TABLE AIMSPC.'|| table_name || ' DELETE STATISTICS;'
  from dba_tab_columns
where last_analyzed is not null
and owner = 'AIMSPC'
;

select distinct 'ANALYZE TABLE COMMON.'|| table_name || ' DELETE STATISTICS;'
  from dba_tab_columns
where last_analyzed is not null
and owner = 'COMMON'
;

select distinct 'ANALYZE TABLE ENTMAGER.'|| table_name || ' DELETE
STATISTICS;'
  from dba_tab_columns
where last_analyzed is not null
and owner = 'ENTMAGER'
;

select distinct 'ANALYZE TABLE OPS$ORACLE.'|| table_name || ' DELETE
STATISTICS;'
  from dba_tab_columns
where last_analyzed is not null
and owner = 'OPS$ORACLE'
;

spool off

set pagesize 30
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
set heading on
set feedback on
set termout on
```

SET_OPTIMIZER.SH

```
#!/usr/bin/ksh
#
# set_optimizer.sh
#
# Parameters
# 1 - instance
#
# Change the optimizer mode to rule for the instance name
# passed as an input parameter. For the change to take effect
# the database must be shutdown and startup again
# This script will change the init.ora file in the following
# directories:
# 1- $ORACLE_HOME/dbs
# 2- /Cots/oracle/admin/<instance_name>/pfile
#
# NOTE: This script does minimal if no error checking - call
#       a driver script to do the error checking
#
args=$#
if [ ${args} -ne 1 ]
then
    echo " "
    echo "Invalid number of arguments"
    echo "Usage: set_optimizer.sh <instance>"
    echo " "
    exit 1
fi
instance=$1

if [ "$ORACLE_HOME" = "" ]
then
    ORACLE_HOME="/Cots/oracle/product/7.3.3"
    export ORACLE_HOME
fi

cp ${ORACLE_HOME}/dbs/init${instance}.ora
${ORACLE_HOME}/dbs/init${instance}.bk
cp ${ORACLE_HOME}/dbs/init${instance}.ora $ORACLE_HOME/dbs/inittemp.ora
#
# take care of case were emacs has edited the init.ora file and
# there is no proper end of file
#
ex $ORACLE_HOME/dbs/inittemp.ora >> /dev/null << XXX
$wq
XXX

ex $ORACLE_HOME/dbs/inittemp.ora >> /dev/null<< XXX
g/optimizer_mode/s/optimizer_mode/#optimizer_mode/g
```

CHANGE PACKAGE CONTROL NUMBER: (F5D-A59-02-02)

```
wq
XXX

ex $ORACLE_HOME/dbs/inittemp.ora >> /dev/null<< XXX
$ 
a

# Set Optimizer mode
optimizer_mode = rule
.
wq
XXX

cp ${ORACLE_HOME}/dbs/inittemp.ora ${ORACLE_HOME}/dbs/init${instance}.ora
rm -f ${ORACLE_HOME}/dbs/inittemp.ora

#
#

PFILE_DIR=/Cots/oracle/admin/${instance}/pfile
export PFILE_DIR

cp ${PFILE_DIR}/init${instance}.ora ${PFILE_DIR}/init${instance}.bk
cp ${PFILE_DIR}/init${instance}.ora ${PFILE_DIR}/inittemp.ora

ex ${PFILE_DIR}/inittemp.ora >> /dev/null << XXX
$ 
wq
XXX

ex ${PFILE_DIR}/inittemp.ora >> /dev/null << XXX
g/optimizer_mode/s/optimizer_mode/#optimizer_mode/g
wq
XXX

ex ${PFILE_DIR}/inittemp.ora >> /dev/null << XXX
$ 
a

# Set Optimizer mode
optimizer_mode = rule
.
wq
XXX

cp ${PFILE_DIR}/inittemp.ora ${PFILE_DIR}/init${instance}.ora
rm -f ${PFILE_DIR}/inittemp.ora
```

TS_MAP.SCR

```
set echo off
set termout off
set feedback off
set pause off
set verify off
set newpage 0
set pagesize 50
set lines 78

/*
   SQL Script to map the area of a tablespace
*/

spool tsmap_&1

ttitle center 'T A B L E S P A C E      M A P' skip 1 -
        center (&1) skip 1 -
        center '=====' right 'PAGE:' -
        format 999 sql.pno skip 2

column segment_name format a30
column segment_name heading 'OBJECT'
column file_id format 999
column file_id heading 'FILE| ID'
column block_id heading 'BLOCK'
column bytes format 999,999,999
column blocks format 999,999,999
column extent_id format 990
column extent_id heading 'ORDER'

select '-- * -- FREE SPACE -- * --' segment_name,
       file_id,
       block_id,
       bytes,
       blocks,
       0 extent_id
  from sys.dba_free_space
 where tablespace_name = upper('&1')
union
select segment_name,
       file_id,
       block_id,
       bytes,
       blocks,
       extent_id
  from sys.dba_extents
 where tablespace_name = upper('&1')
order by 1, 6
/

spool off
ttitle
```